

ЧОВЕШКИЯТ ФАКТОР В СОФТУЕРНОТО ПРОИЗВОДСТВО: СПАСЕНИЕ ИЛИ БЕДСТВИЕ

ст.н.с. д-р Нели Манева

ИМИ – БАН, София, ул. Акад. Г.Бончев, бл.8
nelman@gbg.bg

Резюме

Статията е посветена на актуален проблем – ролята на човешкия фактор в софтуерното производство. Дадени са основни дефиниции, предистория и съвременно състояние на управлението на човешките ресурси в тази област. Представен е накратко авторският подход, включващ формален метод, методология и концепция за интегрирана софтуерна система, подпомагаща прилагането му. Описани са резултатите от използване на подхода за управление на човешките ресурси в СТ. С цел предизвикване на дискусия и/или търсене на решение, са споменати няколко проблема, свързани с подготовката и използването на кадри в разглежданата област.

Ключови думи: управление на човешките ресурси(УЧР), софтуерни технологии, образование и УЧР.

1. Въведение

През 1969г. на конференция на НАТО се обсъждат някои симптоми на „софтуерна“ криза – голям брой некачествени програмни продукти и разочаровани потребители, реализация на софтуерни проекти с надхвърлени ресурси и неизпълнени социални поръчки, недостиг на квалифицирани кадри. Като решение се предлага създаването на нова научно-приложна област - Software Engineering, която на български бе наречена Софтуерни технологии (СТ). Основна цел на СТ е ефективно разработване на качествен софтуер чрез съчетаване на научни подходи с технологични новости, мениджърски похвати и печеливши бизнес-практики. Днес, в условия на световна криза, след повече от 40 години отново се заговори и за софтуерна криза, като се наблюдават същите симптоми. Особено сериозен проблем е недостигът на квалифицирани кадри, които не само не са достатъчни, но и ще намаляват. Според прогнозите, ще има 350 000 свободни работни места за ИТ специалисти в ЕС през 2012г. и около 250 000 - за Индия към 2010г (McKinsey O&O конференция, Виена, 2008 г.). За България предвижданията са за около 15 300 незаети места към 2011г. (БАСКОМ, 2008). Силно намаленият интерес у младите хора към ИТ-специалностите в университетите се дължат на трудно следване и сложна, трудна и стресираща работа (макар и високо платена!), изискваща непрекъснато усвояване на нови знания и умения.

За съжаление, поради глобалната криза стандартните решения като допълнителни инвестиции за наука и иновации не работят. Затова ще потърсим друг подход, възприемайки следната дефиниция за разглежданото направление:

СТ са систематичен подход към разработването на качествен софтуер, предлагането му на пазара, експлоатацията и съпровождането му в рамките на планираните ресурси (финансови, времеви, човешки).

Доколкото финансовите ресурси са недостатъчни, а времевите обикновено зависят от конкретните ограничения на софтуерните проекти, ще разгледаме ролята на човешките ресурси - единствените, които изглеждат многообещаващи поради неизчерпаемостта си, разбира се, при подходящо управление. В Част 2 ще разгледаме особеностите на управлението на човешките ресурси в СТ. В Част 3 ще опишем накратко предлагания от нас подход и получените досега резултати. Четвъртата част ще посветим на някои проблеми при подготовката и използването на кадри за ИКТ, а в Заключениеето ще споменем някои препоръки.

2. Човешкият фактор в Софтуерните технологии

Ролята на хората, участници във всяка дейност е изследвана и оценявана от различни тледни точки и с различни намерения – от задоволяване на любопитството за човешката природа до прилагане на техники за манипулиране, за да се постигнат определени цели. Известна е фразата на Ленин „Кадрите решават всичко“. За СТ показателен е експериментът, проведен със съдействието на IBM в няколко хиляди софтуерни фирми в цял свят, при който на всеки участник се съобщава, че трябва да започне нов проект с двама изпълнители. 98% са попитали кой ще е партньърът им в проекта, а само 2% - какъв е самият проект.

Сериозният научен интерес към проблема се обуславя от факта, че софтуерът се създава от индивиди с определени физически и умствени възможности и различна психологическа нагласа. Изследването и съобразяването с тези личностни характеристики може да подобри организацията на работата и да повиши производителността на труда на разработчиците на софтуер. Освен това софтуерът се използва от хора. Изследването на профила на потребителите, особено във фазите на анализ и проектиране е предпоставка за създаване на надеждни, ефективни и удобни за използване системи. Затова и

традиционната двойка **хардуер-софтуер** напоследък се разширява с нов, трети елемент, наречен **“peopleware”**. Има и повишен интерес към т.н. “софтуерна психология”, създадена от Уейнберг [5]. Тя има за обект на изследване всички участници в процеса на създаване и използване на софтуера. Считаме, че от значение за практиката са най-вече основните теоретични изследвания в две направления:

а) Създаване на психологически профил на софтуерни специалисти – определяне на съвкупност от вродени и/или придобити характеристики, анализирането на които може да подпомага подбора и управлението на софтуерните кадри. Независимо от аргументите за и против използването на психологически профил, много от получените резултати имат и практическа насоченост. Например, при наемане на нови специалисти, често се прилагат тестове за проверка на нивото на двете считани за задължителни качества: адаптивност и устойчивост към стрес.

б) Разработване на методи за въздействие върху хората за постигане на определени цели. Независимо от негативната окраска на тази дейност, която по същество е манипулиране, в основата ѝ са залегнали сериозни теоретични методи и поведенчески модели.

Както вече посочихме, една от основните цели на СТ е ефективен процес на разработване на софтуер. От мениджърска гледна точка, е необходимо да се управляват хората, продукта, процеса и проекта (т.н. четири P's - people, product, process and project). Наредбата не е случайна, защото подценяването на човешкия фактор може да застраши успешната реализация на всеки проект. Наблюдаваното съживяване на интереса към софтуерната психология се доказва от факта, че световно-известният Software Engineering Institute, разработил CMM-модел за глобалната ефективност на дейностите в дадена софтуерна организация чрез определяне на различни нива на зрялост, предлага разширения модел PM-CMM (People Management Capability Maturity Model) [1]. Този модел е създаден с цел “да подобри готовността на софтуерните организации да осъществяват приложения с нарастваща сложност чрез привличане, обучаване, мотивиране и задържане на талантите, необходими за подобряване на бизнеса им за разработване на софтуер”. Определените ключови дейности в модела PM-CMM са наемане на софтуерни специалисти, определяне на структурата и състава на работните групи, управление на взаимоотношенията в работната група, организиране на комуникациите, организиране на сбирки и заседания, работа в екип, и софтуерна ергономика. Последната дейност е професионализъм и етично поведение – създаване и спазване на кодекс на работещите в областта на софтуерната индустрия [4].

3. Нашият подход към управлението на човешките ресурси в СТ

Опитвайки се да намерим подход, който да съответства на съвременните условия и специфичните им особености, започнахме с анализ на текущата обстановка. Преди всичко се налага изводът, че не е възможно да се намираме в постоянна софтуерна криза, защото по дефиниция „криза е повратна точка в развитието на нещо; решителен или критичен момент или етап, след който има подобрение или фатален край”. Необходимо е терминологично уточняване – да не се говори за “софтуерна криза”, а за нещо, което **продължава дълго и/или се повтаря често, непрекъснато създавайки проблеми**. За български еквивалент на въведения от Пресман [3] термин **“chronic affliction”** предлагаме **“хронична недостатъчност”**. Терминът запазва аналогията със съответната медицинска диагноза и отразява точно разминаването между очакванията и реалностите по отношение на софтуера. Недостатъчността в СТ може да бъде количествена (по отношение на измерими параметри като време, пари, автоматизирани средства) или качествена – като неудовлетвореност от постигнатото ниво на по-обща свойства на софтуера (правилност, надеждност, ползваемост и др). Сега дефинираме:

Целта на СТ е да се разработва, предлага и използва софтуерът така, че да се преодолее хроничната недостатъчност и да се удовлетворяват очакванията на обществото.

В съответствие с тази цел разработихме цялостен подход, включващ методология, концепция за софтуерна среда, която я подпомага и процедура за прилагането им. За основа на подхода избрахме следните три основни принципа:

а) Принцип на интерпретацията - да се премине от фактологично описание на обективния свят към диалогичните интерпретации, които прави субектът. За софтуерните технологии това означава да се признава ролята на човешкия фактор и тя да се разглежда като важна и определяща за успешното осъществяване на всяка дейност в областта.

б) Принцип на “разумния избор” - разработването на софтуер да се разглежда като непрекъснат процес на вземане на решения, като се анализират всички възможни алтернативи и се избира най-добрата от тях.

в) Принцип на измерването - повсеместно прилагане на софтуерни метрики, защото както посочва Т. Гилб “Не може да се управлява това, което не може да се измери”.

3.1. Сравнителен анализ и процедура за прилагането му

Формалният математически апарат, който е в основата на подхода, е методът на Сравнителния анализ. Той е представен подробно в [2], а тук ще се спрем само накратко на основните му елементи, за да се добие представа за същността му.

При описание на целите на СТ посочихме, че наред с ефективен процес на разработване, основна цел е и създаването на **качествени** програмни продукти. За съжаление, основен проблем при оценяване на качеството на различните обекти в СТ (продукти, процеси и ресурси) е, че няма абсолютно, а само относително качество. Поставя се въпросът как да дефинираме качеството така, че определението да е конструктивно, т.е. да се свързва с осъществима процедура. Затова под **качество** ще разбираме **степеня на съответствие с предварително зададени изисквания**.

Под **качествено съдържание** на обект ще разбираме качеството, разглеждано в контекста на определена ситуация – т.е с определена цел, от определена гледна точка, с цената на определени ресурси, т.н.

Сравнителният анализ (СА) е изследване на качествено съдържание на еднотипни обекти и съпоставянето им така, че да се избере най-добрият или да се подредят по качество.

Най-общо казано, СА е приложение на теорията на многокритериалното вземане на решения в конкретна област – софтуерните технологии. Осъществяването на СА изисква създаване на **модел на изследвания обект** и описване на **ситуации**, задаващи контекста на прилагането му.

Обектите в СТ могат да продукти, процеси или ресурси. Целта на моделирането на изследвания обект е получаване на количествена мярка на потребителско-ориентираните свойства чрез декомпозирането им на измерими характеристики на едно или няколко нива. Обикновено конструираме йерархичен модел, като корен на дървото е качеството на обекта, а на последното ниво са **метриките**, като за всяко от листата в така построената йерархична структура има процедура за остойностяване.

Друго основно за подхода понятие е понятието **ситуация**.

Под **ситуация (case)** ще разбираме съвкупност от обстоятелства, определящи контекста на прилагане на Сравнителния анализ.

Всяка ситуация **case** се описва чрез наредена петорка:

case = { Goal, Object, Competitors, Task, Level},

в която:

<Goal> задава целта, която може да е изучаване (to characterize), оценяване (to assess), предсказване (to predict), подобряване (to improve), друга – дефинирана от потребителя

<Object> задава обекта на изследване (object of consideration). Обектът може да бъде продукт, процес или ресурс и от произволен вид, например: програмен модул, потребителска документация, спецификация, метод, модел, софтуерно средство и т.н.

<Competitors> задава множество на сравнявани обекти от избрания вид (target objects).

<Task> - може да бъде **избиране** (selection) на най-добрия обект, **ранжиране** (ranking) - получаване на нареден списък на сравняваните обекти или **класифициране** (classification) на обектите към няколко предварително зададени класове на качеството.

<Level> задава ниво на изчерпателност на провеждания анализ (depth level) в зависимост от важността на решавания проблем и отделените за него ресурси.

За реализацията на подхода са необходими две специални длъжностни позиции в софтуерната организация: **Аналитик** и **Клиент**, за които са определени изискваната квалификация, опит и професионални умения за успешно справяне със задълженията.

За управление на човешките ресурси в СТ досега са конструирани модели на обект „участник в разработването на софтуер“ за позициите разработчик, тестер, ръководител на проект, софтуерен инженер, отговорник за осигуряване на качеството и ръководител на софтуерна фирма. Създадените модели са приложени в ситуации за избиране на най-подходящия кандидат за определена позиция (измежду описаните по-горе), за атестиране на служители, за определяне на най-добър служител на годината и др. Получаваните чрез Сравнителния анализ резултати са сравнявани с независимото оценяване, извършвано от назначаваните комисии, като при разминавания в крайния резултат са обсъждани възможните причини, така че да се усъвършенства методът на СА чрез прецизиране на настроаемите параметри – скали за оценка, теглови коефициенти и др.

3.2. Методология INSPIRE и софтуерна система, подпомагаща прилагането ѝ

За систематично и целенасочено прилагане на Сравнителния анализ е разработена и методология, наречена INSPIRE (вдъхновявам). Името е избрано така, че подсказва предназначението на методологията като улесняваща потребителите и е ахроним от важни характеристики на подхода: (Incremental, Neat, Scalable, Permanent, Integrated, Right и Estimable). Засега предложената методология е успешно приложена за пет дейности в СТ: осигуряване на качеството, разработване на софтуер с подизпълнители (outsourcing), осигуряване на ползваемост, оценяване на модели в СТ и управление на човешките ресурси. Както се вижда, избраните дейности са с различен обхват, сложност и значимост за СТ, за да могат да се проверят описаните по-горе характеристики на подхода. Разработена е и концепция за интегрирана софтуерна среда, поддържаща методологията.

3.3. Прилагане на подхода за управление на човешките ресурси в СТ

Като илюстрация на използването на добри практики, ще формулираме 8 прости правила, спазването на които би усъвършенствало дейностите по управление на човешките ресурси. Както ще се види, правилата са с различен обхват на приложимост: първите три лесно се модифицират за каква да е дейност, докато останалите пет са специфични за дейността управление на персонала.

I. Избиране и назначаване на “качествен” мениджър човешки ресурси. Той създава стратегическа (дългосрочна, обикновено 3-годишна) програма и тактически (краткосрочни) планове за управление на човешките ресурси. Мениджърът създава и “профил” на всеки служител на фирмата. В задълженията му влиза и консултиране на всеки по чисто “човешки” проблеми. Практиката показва, че точно тази функция на психотерапевт и душеприказчик води до създаване на подходящ психологически климат, а „разтоварването“ при споделяне на лични проблеми може да увеличи значително производителността на труда.

II. Прилагане на ППП-принципа, според който осъществяването на подхода за УЧР трябва да става:

- Постепенно – инкрементална (постепенно разширяваща се) реализация на всички програми, свързани с УЧР, с междинно оценяване и настройка;
- Постоянно – не еднократни “ad hoc” инициативи, а следване на дългосрочна и добре обмислена стратегия;
- Правилно – прилагане на доказали полезността си добри практики.

III. Осигуряване на продължаваща (life-long) квалификация и обучение

Препоръката е за разделяне на служителите в групи и създаване на програми за кариерно развитие на всяка група, в съответствие с нейните цели и задачи. Определя се подходящо съдържание на задължителното професионално знание (body of knowledge), набора от специфични техники и умения, както и естеството и присъщите видове комуникации за всяка група.

4. Реализация на индивидуален подход - съобразяване с индивидуалния профил и ценностната система на всеки служител. Препоръчва се предварителна подготовка при срещите със служителите, доколкото е доказано, че ефективна комуникация е възможна само ако двете страни са на едно и също ниво. Интересен аспект на приложната психология, който има практическо значение, са техниките за оценка и нагласяне към нивото на събеседника/партньора).

5. Отчитане на факторите, определящи персоналната мотивация като главна движеща сила.

6. Съобразяване с т.н. „когнитивен дисонанс“ - всеки смята, че постига успехите си сам, а за провалите са виновни другите!

7. Работа в екип – съвременните проекти са големи и сложни и могат да се реализират успешно само с колективните усилия на екиписти. Неформалното описание на колективния труд е: “Всички сме ангели с едно крило и можем да полетим само ако се опрем един на друг”. Приложната психология препоръчва обучение за работа в екип, доколкото са необходими специфични знания и умения, които трябва да бъдат усвоени чрез различни тематични семинари и тренинги.

8. Въздръжане от критикуване на служител, особено в присъствието и на други лица. Твърдението на специалистите е, че няма гравивна критика – отправянето на обвинения обикновено засяга достойнството на критикувания, поражда психически дискомфорт и желание за отмъщение.

Не може да се очаква, че такава сложна и отговорна дейност като управлението на човешките ресурси може да се опише и осъществи ефективно само чрез спазване на тези 8 правила, но те могат да се разглеждат като добра стартова основа за професионален подход към човешкия фактор, който би допринесъл за повишаване на производителността на труда и за успешно функциониране на софтуерната организация като цяло.

4. Проблеми на подготовката и използването на кадри за ИКТ

Както вече споменахме, има не само недостиг на квалифицирани кадри в ИКТ, но и прогнозите са за по-нататъшното им намаляване. Анализът на причините за тази световна тенденция е предмет на сериозни изследвания. Считаме, че такъв анализ би трябвало да се прави задълбочено, въз основа на големи съвкупности от достоверни и систематично събирани данни, с отчитане на специфичните условия във всяка страна. Затова и ние си поставяме далеч по-скромната задача - да споделим свои наблюдения по някои проблеми с надеждата, че ще предизвикаме обсъждане и може би и решаването им на съответното ниво на компетентност..

Преди всичко ще отбележим, че проблемът е разпознат като съществен в рамките на Европейския съюз. В разработената стратегия Европа 2020 е обърнато специално внимание на страните от ЕС, че трябва да направят всичко възможно и “да гарантират достатъчна наличност на кадри с природонаучно, математическо и инженерно образование и да фокусират училищните програми върху изграждане на творческо, иновативно и предприемаческо мислене”. Според нас тази директива на ЕС, поставяща изисквания към училищните курсове, е перспективна, защото позволява ранно “зарибяване” на децата.

Така интересът към компютрите, породен от компютърните игри, може да бъде поддържан чрез по-нататъшни занимания с програмиране. Резултатите от извънкласната работа по информатика показват, че дори ученици от 3 до 5 кл. успешно се справят с усвояването на основни алгоритми и написването на програми за тях. За съжаление, вниманието в българското училище е насочено предимно към ИТ - за овладяване на основни компютърни умения и работа с няколко полезни софтуерни пакети. Все още няма адекватни учебни програми по информатика, а именно те биха били основа за подготовката на бъдещите софтуерни разработчици.

Университетското образование в информатичните специалности се провежда по добри учебни програми, но като цяло оценките за качеството на подготовката не са високи. Причините могат да се търсят както в процеса на организиране на обучението (недостиг на квалифицирани преподаватели, висока лекционна натовареност и липса на време за научно-изследователска работа и осъвременяване на курсовете, борба за привличане и задържане на студенти на всяка цена поради обвързване на финансирането с броя на обучаваните студенти), така и в отношението на повечето студенти (липса на желание за посещаване на учебни занятия, липса на инициативност и интереси за изява в научно-приложни разработки и др.)

Друг проблем, който искаме да споменем, са странните отношения между организациите, които отговарят за подготовката или наемат специалисти за работа в ИКТ-сектора. Свидетели сме на непрекъснатите пререкания между представители на бизнеса и университетите. Бизнесът е недоволен от ниското ниво на знания и професионална квалификация на завършващите бакалаври и магистри, а университетите се оплакват от нежеланието на фирмите да участват (с кадрови и финансови ресурси) както в обучението, така и в съвместни научно-приложни проекти. Едва в последните години няколко водещи софтуерни фирми се престрашиха да започнат изследователска дейност, като стартираха съвместни иновационни проекти с участието на специалисти от университети или научни институти. Включването на представители на софтуерния бизнес като консултанти в научни проекти, поканени лектори или ръководители на дипломни работи изглежда обещаващо и би улеснило както трансфера на научни постижения в практиката, така и успешно кариерно развитие на младите специалисти.

5. Заключение

В статията са разгледани някои основни проблеми на управление на човешките ресурси, като е предложен и формален метод, доказал полезността си при реализация на дейности в СТ, свързани с човешкия фактор. Считаме, че наред със създаване на нагласи за важноста на хората като единствен неизчерпаем ресурс, най-ефективни биха били:

- планиране и осъществяване на програми за личностно развитие във всички степени на образованието;
- съчетаване на изследователската работа в науката със създаването и на процедури за трансфер на резултатите в практиката;
- инициране на взаимно-изгодни съвместни проекти между университети, научни звена и софтуерни фирми, така че последните да се „отворят“ (и познавателно, и финансово) за иновации.

Направените конкретни препоръки могат да бъдат полезни както за мениджърите на различно ниво в софтуерните организации, така и за членовете на професионалната ни общност от научни работници и преподаватели по информатика, информационни и софтуерни технологии. Предлагаият подход подпомага локалната оптимизация. Постигането ѝ в личностен план означава всеки от нас да се стреми да изпълнява задълженията си съвестно и на най-високото възможно професионално ниво. Работещ ли е този подход? Моят отговор е: за тактически цели (в отделна организация или отрасъл) – **ДА**, но за стратегически цели (особено на национално ниво) – **НЕ!** В това всеки може да се убеди, като се запознае със съдържанието на разработените Национални стратегии за наука, за образование или за развитие на ИКТ. Важният въпрос е **накъде и как** продължаваме. Нека всеки сам определи отговор(ност)а си, съзнавайки, че **Светът (на компютрите) е голям и спасение (бедствие) дебне отвсякъде!**

Литература

1. Curtis, B. et al, People Management Capability Maturity Model for Software, SEI, Carnegie Mellon University, Pittsburgh, PA, 1993.
2. Maneva N. Comparative Analysis: A Feasible Software Engineering Method. *Serdica J. of Computing*, vol.1, N=1, 2007, pp.1-12.
3. Pressman R. *Software engineering – A Practitioner’s Approach*, Sixth edition, McGraw Hill, 2005.
4. Thomseff R. *Crossing the line: Professionalism and Software Ethics*. *American Programmer*, vol.6, No 7, 1993.
5. Weinberg G.M., *The Psychology of Computer Programming*. Van Nostrand Reinhold, 1971